



# WebServices API Documentation

Last Updated 16<sup>th</sup> May 2011



First Option Solutions Ltd  
Signal House  
Jacklyns Lane  
Alesford  
Hampshire  
SO24 9JJ

## **IssueCentre WebServices API**

### **Software Release: 2.4.1.16471**

First Option Solutions Ltd  
Revised 16<sup>th</sup> May 2011  
Copyright © 2009 First Option Solutions.

This book may be copied, printed, and distributed; however, attribution must be given to First Option Solutions Ltd. This book cannot be modified or used for commercial purposes without prior permission from First Option Solutions.

First Option Solutions Ltd assumes no responsibility for errors or omissions or damages resulting from the use of the information contained herein.

---

# TABLE OF CONTENTS

1. Introduction.....	5
1.1. Typography Conventions.....	5
1.2. Glossary of Terms.....	6
2. Connection WebService .....	7
2.1. getContracts Function .....	7
2.2. generateKey Function.....	8
3. Customer WebService .....	9
3.1. getCompanies Function.....	9
3.2. getContacts Function.....	10
3.3. createContact Function .....	11
3.4. createCompany Function .....	12
3.5. updateContact Function.....	13
3.6. updateCompany Function .....	14
4. Contract WebService .....	15
4.1. getPriorities Function .....	15
4.2. getErrorTypes Function .....	16
4.3. getStatusTypes Function .....	17
4.4. getVendors Function .....	18
4.5. getProducts Function .....	19
4.6. getEventTypes Function .....	20
4.7. getReceivedByMethods Function.....	21
4.8. geSummaryDetails.....	21
5. Ticket WebService .....	23
5.1. addEventToTicket Function .....	23
5.2. createTicket Function .....	24

5.3.	uploadFileToTicket Function .....	26
5.4.	searchTickets Function .....	27
5.5.	addReminderToTicket Function .....	29
5.6.	toggleNotificationForTicket Function.....	30
5.7.	sendEmailResponseFromTicket Function.....	31
5.8.	getEventsForTicket .....	32
5.9.	getAllTags .....	32
5.10.	getClosedTickets.....	32
5.11.	getOpenTickets.....	32
5.12.	getTicketFixGroup .....	32
5.13.	setTicketFixGroup.....	32

---

# INTRODUCTION

This document provides the details of the API that can be used for integrating with IssueCentre. It provides access to the core IssueCentre functionality of Creating, Updating and Searching tickets. The API is provided as a series of WebService functions that can be accessed by HTTP or securely by HTTPS according to the installation configuration. The document is targeted at web developers that have knowledge of calling WebServices.

Please contact First Option Solutions on +44 (0)333 900 1123 if you require assistance in implementing the WebService calls and for sample code.

Please see the other IssueCentre documentation for further reference:

- Installation Guide
- User Guide
- Administrator's Guide

---

## 1.1. Typography Conventions

Formatting conventions are used through the document with the following meanings.

*Italics*

Button names are formatted in italics

**Bold**

Main menu items are formatted in bold type

Highlighted Section

Useful tips and explanations are highlighted in

---

## 1.2. Glossary of Terms

The following list describes the meaning of terms used throughout this document.

<b>Term</b>	<b>Description</b>
API	Application Programmers Interface
WebService	A standard means to call a function on the IssueCentre server
APIKey	An IssueCentre value provided by the API to identify the user that the application is connecting as.
Ticket	The term used to reference an issue that is to be tracked and resolved.
Event	An action that takes place that has an effect on a ticket. Events may be caused by human intervention, e.g. an agent logging a follow-up telephone call, or by IssueCentre system monitoring e.g. the SLA for the ticket being breached.
Agent	A user that is assigned the task of creating and resolving ticket issues.
Contract	A complete segregation of tickets and configuration that apply to a separate customer contract.
Knowledgebase	A structured collection of FAQs that can be used to resolved commonly occurring ticket issues.
SLA	Service Level Agreement. IssueCentre uses SLAs to ensure tickets are processed within the required timescales.
Record Locking	A means to ensure that two agents do not work on the same ticket/email at the same time
Spam	Unwanted/unsolicited email that is of no commercial value

---

## 2. CONNECTION WEBSERVICE

The Connection webservice functions provide the means to connect to the IssueCentre API. The generateKey function must be used before any of the other WebServices can be used.

---

### 2.1. getContracts Function

Provides a list of IssueCentre contracts that the user is permitted to access using the web application or API.

**Definition**                      **Connection.getContracts (String username, String password)**

**Parameters**

**String username**              The Username of the user to get the list of contracts for

**String password**              The Password of the user to get the list of contracts for

**Returns**                          An XML string containing the list contracts the user is permitted to access.

**Sample Response**

```
<?xml version="1.0" encoding="UTF-8"?>
<return error="0" >
<contracts defaultContract="300">
  <contract id="300"><name>First Option Solutions</name></contract>
  <contract id="11"><name>Demonstration</name></contract>
  <contract id="133"><name>Acme Corporation</name></contract>
</contracts>
</return>
```

**Implementation Notes**

The contract id codes returned in the XML are required for use in the generateKey function described below.

---

## 2.2. generateKey Function

This function generates a key that is used for every other webservice call. The key request is akin to logging into the site. It takes a username, password and contract as parameters and returns an encoded, time limited string.

**Definition**                      **Connection.generateKey(String username, String password, int contract)**

### Parameters

**String username**            The Username of the user to login

**String password**            The Password of the user to login

**int contract**                 The Contract code for which functions are going to be performed

**Returns**                        A string containing an encoded set of authentication details used for all the other API calls

**Sample Response**            <?xml version="1.0" encoding="UTF-8"?>  
                                 <return error="0">  
                                 -4ec7f6a4b45afcf81e5b1a5e478618c65d4396cf0afe052af616318777aa3499fb3f949aca1  
                                 </return>

### **Implementation Notes**

The key is valid for the period of *1 hour* from the time of generation. After this time the function calls will return an error. If required, the application should set a timer to refresh the key at suitable intervals or alternatively refresh the key upon detecting the error.

Multiple keys can be retrieved and used if an application needs to manipulate multiple contracts, but the application will need to ensure they use the correct key accordingly.



---

## 3. CUSTOMER WEBSERVICE

The Customer Webservice provides functions to access and update the details of the IssueCentre customers and their contacts.

---

### 3.1. getCompanies Function

Provides a means to obtain the list of companies that are held in IssueCentre that tickets can be created against. The list will be restricted to the contract specified by the APIKey.

**Definition** `Customer.getCompanies(String apiKey, String wildcard)`

**Parameters**

- String apiKey** The API identifier key used to identify the user and contract
- String wildcard** The wildcard used to match the company names. E.g. a\*, Brit?sh. The wildcard is case insensitive. An empty string will return all companies.

**Returns** An XML string containing the list of company details that match those requested.

**Sample Response**

```
<?xml version="1.0" encoding="UTF-8"?>
<return error="0">
<companies>
  <company id="300001" notifyoptout="yes"><name>First Option Solutions
Ltd</name><tel>01962738380</tel><address></address><country
id="2">England</country><ref>abc001</ref></company>
</companies>
</return>
```

**Implementation Notes**

The function returns a list of Companies which match the specified wildcard-based search string.

---

## 3.2. getContacts Function

Returns a list of contacts for a specified company or all contract companies

**Definition** `Customer.getContacts(String apiKey, int company, String wildcard)`

**Parameters**

- String apiKey** The API identifier key used to identify the user and contract
- int company** The company code to limit the contacts to, 0 will return all contacts across the entire contract
- String wildcard** The wildcard used to match the contact names. E.g. a\*, AI?n. The wildcard is case insensitive. An empty string will return all contacts for the specified company.

**Returns** An XML string containing the list of contacts that match those requested

**Sample Response**

```
<?xml version="1.0" encoding="UTF-8"?>
<return error="0">
<contacts>
  <contact id="12" companyId="300001" primary="yes" notifyoptout="no"><name>Chris
White</name><email>czw@fosolutions.co.uk </email><company>First Option
Solutions</company><tel>01962738241</tel><mob>0777738241</mob></contact>
</contacts>
</return>
```

**Implementation Notes**

Passing a Company value of 0 (zero) will return the full list of contracts for the Contract specified by the APIKey.

---

### 3.3. createContact Function

Creates a contact record against which a ticket can be created.

<b>Definition</b>	<b>Customer.createContact(String apiKey, int company, String surname, String forename, String email, String telephone, String mobile, boolean notificationDisabled)</b>
<b>Parameters</b>	
<b>String apiKey</b>	The API identifier key used to identify the user and contract
<b>int company</b>	The company to create the contact within.
<b>String surname</b>	The surname of the contact.
<b>String forename</b>	The forename of the contact.
<b>String email</b>	The email address of the contact.
<b>String telephone</b>	The telephone number of the contact.
<b>String mobile</b>	The mobile number of the contact.
<b>boolean notificationDisabled</b>	The notification status for the contact (true if they should be disabled)
<b>Returns</b>	On success, the integer code of the newly created contact.
<b>Sample Response</b>	<pre>&lt;?xml version="1.0" encoding="UTF-8"?&gt; &lt;return error="0"&gt;306091&lt;/return&gt;</pre>

#### ***Implementation Notes***

Disabling the notifications prevents the contact from receiving any further ticket update notifications. The Company value can be obtained, either by creating a new Company using the createCompany function or by obtaining an existing company using the getCompanies function.

---

## 3.4. createCompany Function

Creates a company record against which contacts are stored.

<b>Definition</b>	<b>Customer.createCompany(String apiKey, int country, boolean notificationDisabled, String address, String fax, String name, String notes, String telephone)</b>
<b>Parameters</b>	
<b>String apiKey</b>	The API identifier key used to identify the user and contract
<b>int country</b>	The country for the company
<b>boolean notificationDisabled</b>	The notification status for the company (true if they should be disabled)
<b>String address</b>	The address for the company
<b>String fax</b>	The fax number for the company.
<b>String name</b>	The name for the company.
<b>String notes</b>	Any notes about the company
<b>String telephone</b>	The telephone number for the company
<b>Returns</b>	On success, the integer of the newly created company.
<b>Sample Response</b>	<pre>&lt;?xml version="1.0" encoding="UTF-8"?&gt; &lt;return error="0"&gt;300016&lt;/return&gt;</pre>

### ***Implementation Notes***

Disabling the notifications prevents any contacts of the company from receiving any further ticket update notifications. The Country value can be obtained, by using the getCountries function.

---

## 3.5. updateContact Function

Updates a company contact using the specified data.

<b>Definition</b>	<b>Customer.updateContact(String apiKey, int company, String surname, String forename, String email, String telephone, String mobile, boolean notificationDisabled, int contact)</b>
<b>Parameters</b>	
<b>String apiKey</b>	The API identifier key used to identify the user and contract
<b>int company</b>	The company to update the contact with.
<b>String surname</b>	The surname of the contact.
<b>String forename</b>	The forename of the contact.
<b>String email</b>	The email address of the contact.
<b>String telephone</b>	The telephone number of the contact.
<b>String mobile</b>	The mobile number of the contact.
<b>boolean notificationDisabled</b>	The notification status for the contact (true if they should be disabled)
<b>int contact</b>	The contact that should be updated
<b>Returns</b>	On success, the integer of the updated contact.
<b>Sample Response</b>	<pre>&lt;?xml version="1.0" encoding="UTF-8"?&gt; &lt;return error="0"&gt;306091&lt;/return&gt;</pre>

### ***Implementation Notes***

Disabling the notifications prevents the contact from receiving any further ticket update notifications.

## 3.6. updateCompany Function

Updates a company using the specified data

<b>Definition</b>	<b>Customer.updateCompany(String apiKey, int country, boolean notificationDisabled, String address, String fax, String name, String notes, String telephone, int company)</b>
<b>Parameters</b>	
<b>String apiKey</b>	The API identifier key used to identify the user and contract
<b>int country</b>	The country for the company
<b>boolean notificationDisabled</b>	The notification status for the company (true if they should be disabled)
<b>String address</b>	The address for the company
<b>String fax</b>	The fax number for the company.
<b>String name</b>	The name for the company.
<b>String notes</b>	Any notes about the company
<b>String telephone</b>	The telephone number for the company
<b>int company</b>	The company that should be updated
<b>Returns</b>	On success, the integer of the updated company.
<b>Sample Response</b>	<pre>&lt;?xml version="1.0" encoding="UTF-8"?&gt; &lt;return error="0"&gt;300016&lt;/return&gt;</pre>

### **Implementation Notes**

Disabling the notifications prevents any contacts of the company from receiving any further ticket update notifications. The Country can be obtained, by using the getCountries function.

---

## 4. CONTRACT WEBSERVICE

The Contract Webservice functions provide the system values required to create tickets and companies. The values provided by these functions can be administered using the IssueCentre Web application.

---

### 4.1. getPriorities Function

Returns a list of priorities in order of level. Priorities are assigned to tickets to classify their urgency and importance. Priorities are typically called P1-P5 but can be named as required.

<b>Definition</b>	<b>Contract.getPriorities(String apiKey)</b>
<b>Parameters</b>	
<b>String apiKey</b>	The API identifier key used to identify the user and contract
<b>Returns</b>	An XML string containing the list priorities for the contract.
<b>Sample Response</b>	<pre>&lt;?xml version="1.0" encoding="UTF-8"?&gt; &lt;return error="0"&gt; &lt;priorities&gt;   &lt;priority id="300000" default="0" iLevel="0"&gt;     &lt;name&gt;P1&lt;/name&gt;   &lt;/priority&gt;   &lt;priority id="300008" default="0" iLevel="1"&gt;     &lt;name&gt;P2&lt;/name&gt;   &lt;/priority&gt;   &lt;priority id="300001" default="0" iLevel="2"&gt;     &lt;name&gt;P3&lt;/name&gt;   &lt;/priority&gt;   &lt;priority id="300004" default="0" iLevel="3"&gt;     &lt;name&gt;P4&lt;/name&gt;   &lt;/priority&gt; &lt;/priorities&gt; &lt;/return&gt;</pre>

---

## 4.2. getErrorTypes Function

Error Types are used to classify the type of issue that is being raised by the ticket. This function returns a list of the currently available error types.

**Definition**                      **Contract.getErrorTypes (String apiKey)**

**Parameters**

**String apiKey**                      The API identifier key used to identify the user and contract

**Returns**                              An XML string containing the list error types for the contract.

**Sample Response**

```
<?xml version="1.0" encoding="UTF-8"?>
<return error="0">
<errortypes>
  <errortype id="114"><name>Server Slow</name></errortype>
  <errortype id="121"><name>Server Down</name></errortype>
  <errortype id="122"><name>Server Error</name></errortype>
  <errortype id="124"><name>User Error</name></errortype>
  <errortype id="1116"><name>General Enquiry</name></errortype>
  <errortype id="1131"><name>Connectivity</name></errortype>
  <errortype id="1142"><name>Basic Setup</name></errortype>
</errortypes>
</return>
```



## 4.3. getStatusTypes Function

Statuses are used to classify the progress of the lifecycle of a ticket. This function returns a list of status types that are used to classify the status of a ticket.

**Definition** **Contract.getStatusTypes (String apiKey)**

### Parameters

**String apiKey** The API identifier key used to identify the user and contract

**Returns** An XML string containing the list status types for the contract.

**Sample Response**

```
<?xml version="1.0" encoding="UTF-8"?>
<return error="0">
<statustypes>
  <statustype id="300000" closed="0" default="0" escalate="0" suspendSLA="0">
    <name>Open</name>
  </statustype>
  <statustype id="300001" closed="0" default="0" escalate="0" suspendSLA="0">
    <name>Closed</name>
  </statustype>
  <statustype id="300002" closed="0" default="0" escalate="0" suspendSLA="0">
    <name>Pending</name>
  </statustype>
  <statustype id="300004" closed="0" default="0" escalate="0" suspendSLA="0">
    <name>Suspended</name>
  </statustype>
  </statustype>
  <statustype id="300006" closed="0" default="0" escalate="0" suspendSLA="0">
    <name>Info</name>
  </statustype>
  <statustype id="300033" closed="0" default="0" escalate="0" suspendSLA="0">
    <name>In Dev</name>
  </statustype>
  <statustype id="300071" closed="0" default="0" escalate="0" suspendSLA="0">
    <name>Awaiting Testing</name>
  </statustype>
</statustypes>
</return>
```

### Implementation Notes

The Closed attribute indicates whether the ticket is closed. The default attribute indicates whether the status has been marked as the default status to be used for new tickets. The escalate attribute indicates whether the status indicates that the ticket has been escalated to another vendor/agency/3<sup>rd</sup> line support.

The suspendSLA attribute indicates whether the SLA monitoring is suspended for tickets set to this status. This is useful for example, if the contact is unavailable for a known period.

---

## 4.4. getVendors Function

This function obtains the list of vendors that supply the products that are being supported and for which tickets can be raised against.

**Definition**                    **Contract.getVendors(String apiKey)**

**Parameters**

**String apiKey**                The API identifier key used to identify the user and contract

**Returns**                        An XML string containing the list status types for the contract.

**Sample Response**

```
<?xml version="1.0" encoding="UTF-8"?>
<return error="0">
<vendors>
<vendor id="1024"><name>Microsoft</name></vendor>
</vendors>
</return>
```

## 4.5. getProducts Function

This function retrieves the list of products that tickets can be raised against.

**Definition**                      **Contract.getProducts (String apiKey)**

**Parameters**

**String apiKey**                      The API identifier key used to identify the user and contract

**Returns**                              An XML string containing the products, and their associated versions, platforms and platformversions, for the contract encoded in the APIKey

**Sample Response**                      `<?xml version="1.0" encoding="UTF-8"?>`  
`<return error="0">`  
`<products>`  
`<product id="10058" vendor="1075">`  
`<name>Office 2008</name>`  
`<versions>`  
`<version productversionId="10047" platformversionId="174">`  
`<name>1.0</name>`  
`<platform>Windows</platform>`  
`<platformversion>XP</platformversion>`  
`</version>`  
`</versions>`  
`</product>`  
`<product id="10055" vendor="1075">`  
`<name>Office 2007</name>`  
`<versions>`  
`<version productversionId="10047" platformversionId="174">`  
`<name>1.0</name>`  
`<platform>Windows</platform>`  
`<platformversion>XP</platformVersion>`  
`</version>`  
`<version productversionid="10067" platformversionid="174">`  
`<name>2.0</name >`  
`<platform>Windows</platform>`  
`<platformversion>XP</platformversion>`  
`</version>`  
`</versions>`  
`</product>`  
`</products>`  
`</return>`

---

## 4.6. getEventTypes Function

This function retrieves a list of the types of events that can be added to tickets.

**Definition**                    **Contract.getEventTypes (String apiKey)**

**Parameters**

**String apiKey**                The API identifier key used to identify the user and contract

**Returns**                        An XML string containing the list event types for the contract.

**Sample Response**

```
<?xml version="1.0" encoding="UTF-8"?>
<return error="0">
<eventtypes>
  < eventType id="300000">
    <name>Emailed Contact</name>
  </eventType>
  <eventType id="300001">
    <name>Telephoned Contact </name>
  </eventType>
</eventTypes>
</return>
```

---

## 4.7. getReceivedByMethods Function

This function obtains a list of the options that can be used to classify new tickets as to the method that was used by the contact to raise the ticket issue.

**Definition**                    **Contract.getReceivedByMethods (String apiKey)**

**Parameters**

**String apiKey**                The API identifier key used to identify the user and contract

**Returns**                        An XML string containing the list of received by methods for the contract.

**Sample Response**

```
<?xml version="1.0" encoding="UTF-8"?>
<return error="0">
<receivedbys>
  <receivedby id="300000">Received By API</receivedby>
  <receivedby id="300001">Received By Telephone</receivedby>
  <receivedby id="300002">Received By Email</receivedby>
</receivedbys>
</return>
```

---

## 4.8. getSummaryDetails Function

This function obtains dashboard values for the contract. These values include total open tickets, total closed tickets, total open tickets for current user, date of last ticket update and total open tickets with highest priority

<b>Definition</b>	<b>Contract.getSummaryDetails (String apiKey</b>
<b>Parameters</b>	
<b>String apiKey</b>	<b>The API identifier key used to identify the user and contract</b>
<b>Returns</b>	<b>An XML string containing the dashboard data for the contract.</b>
<b>Sample Response</b>	<pre>&lt;?xml version="1.0" encoding="UTF-8"?&gt; &lt;return error="0"&gt; &lt;summary&gt;   &lt;open&gt;24&lt;/open&gt;   &lt;closed&gt;178&lt;/closed&gt;   &lt;agentsopen&gt;7&lt;/agentsopen&gt;   &lt;toppriority&gt;4&lt;/toppriority&gt;   &lt;lastupdated&gt;7&lt;/lastupdated&gt; &lt;/summary&gt; &lt;/return&gt;</pre>

---

## 5. TICKET WEBSERVICE

The Ticket WebService functions provide the core functions to create and update IssueCentre tickets.

---

### 5.1. addEventToTicket Function

This function provides the means to record a new piece of activity that has taken place on an existing ticket.

**Definition** `Ticket.addEventToTicket (String apiKey, int ticket, String eventText, int statusType, int eventType, int timeSpent)`

**Parameters**

<b>String apiKey</b>	The API identifier key used to identify the user and contract
<b>int ticket</b>	The ticket code to add the event to.
<b>int eventText</b>	The text to be recorded for the event
<b>int statusType</b>	The status type code that the ticket should be set to upon adding the even
<b>int eventType</b>	The event type that this event should be added as
<b>int timeSpent</b>	The agent time spent on this event in seconds

**Returns** A String "Success" or another string representing the error that occurred

**Sample Response**

```
<?xml version="1.0" encoding="UTF-8"?>
<return error="0">Success</return>
```

**Implementation Notes**

The statusType value should be a value from those provided by the getStatusTypes function. The EventType value should be a value from those provided by the getEventTypes function.

## 5.2. createTicket Function

This function creates a new IssueCentre ticket.

**Definition** `TicketService.createTicket(String apiKey, int company, int contact, int product, int productVersion, int ticketStatus, int priority, String supplierRef, String clientRef, String internalNotes, int platform, int platformVersion, int errorType, int receivedBy, String issue, int timeSpent, String summary, String tagList, int iFixGroup)`

### Parameters

<b>String apiKey</b>	The API identifier key used to identify the user and contract
<b>int company</b>	The company which this ticket applies
<b>int contact</b>	The contact the ticket will be created on behalf of
<b>int product</b>	The product code for the product the ticket applies to
<b>int productVersion</b>	The product version code for the product the ticket applies to
<b>int ticketStatus</b>	The status to set the ticket to upon creation
<b>int priority</b>	The priority of the ticket
<b>String supplierRef</b>	The supplier reference to assign to the ticket
<b>String clientRef</b>	The client reference to assign to the ticket
<b>String internalNotes</b>	The Internal notes to add to the ticket
<b>int platform</b>	The Platform the ticket applies to
<b>int platformVersion</b>	The Platform version the ticket applies to
<b>int errorType</b>	The Error Type that should be set on the ticket
<b>int receivedBy</b>	The Received By that should state how this ticket was received
<b>String Issue</b>	The description of the problem for the ticket
<b>int timeSpent</b>	The number of seconds spent creating the ticket
<b>String summary</b>	This summary of the ticket
<b>String tagList</b>	A comma separated list of tags for the ticket
<b>int iFixGroup</b>	The Fix Group that the ticket belongs to

**Returns** The code of the ticket created or negative values for an error

**Sample Response**

```
<?xml version="1.0" encoding="UTF-8"?>
<return error="0">300078</return>
```



### **Implementation Notes**

The company and contact values are obtained using the GetCompanies, GetContacts or CreateCompany, Create Contact functions respectively. The product and productVersion values are obtained using the getProducts function. The ticketStatus and priority values are obtained using the getStatusTypes and getPriorities functions. The platform and platformVersion values are also obtained from the getProducts function. The errorType and receivedBy values are obtained from the getErrorTypes and getReceivedBy functions. The tagList values are obtained from the getAllTags functions

The values that are mandatory are **apiKey, company, contact, product, ticketStatus, priority, errorType, receivedBy, issue, timeSpent** and **summary**. All other fields may have null values passed through for them.

The values that are passed through to **tagList** should be the names of the tags that apply to the ticket in a comma separated list, I.E. "Password Reset, New User, Important".

## 5.3. uploadFileToTicket Function

This function allows a file to be associated with an existing ticket. The file is retrieved from an FTP server.

<b>Definition</b>	<b>Ticket.uploadFileToTicket(String apiKey, int ticket, boolean deleteAfterUse, String fileName, String connectionString, String path, String ftpUser, String ftpPass)</b>
<b>Parameters</b>	
<b>String apiKey</b>	The API identifier key used to identify the user and contract
<b>int ticket</b>	The ticket to add the file to.
<b>boolean deleteAfterUse</b>	Instructs the server to delete the file after downloading it.
<b>String fileName</b>	The filename of the file to be added into the ticket
<b>String connectionString</b>	The connection String used to connect to the FTP Server to download the file. If "sftp://" is at the beginning of the connection string then the file will be downloaded using SFTP rather than FTP, providing the FTP server supports SFTP
<b>String path</b>	Relative path from login location to get to the directory containing the file.
<b>String ftpUser</b>	The ftp user which uploaded the file
<b>String ftpPass</b>	The password for the ftp user.
<b>Returns</b>	The String "Success" or another string representing the error that occurred
<b>Sample Response</b>	<pre>&lt;?xml version="1.0" encoding="UTF-8"?&gt; &lt;return error="0"&gt;Success&lt;/return&gt;</pre>

### **Implementation Notes**

This function must be made once the file is available on the appropriate server in the specified location. IssueCentre will retrieve the file from the server using the connection details, path and user details provided. If the FTP server is on a different machine to IssueCentre it is recommended that sftp is used to ensure the data transfer is encrypted. The FTP server must be configured to ensure the provided FTPUser has permission to access and download the specified file.

The DeleteAfterUse parameter indicates whether the file is to be deleted after the file has been successfully added to the ticket. The file will not be deleted if an error has occurred.

---

## 5.4. searchTickets Function

This function provides the means to search the IssueCentre database.

**Definition** `TicketService.searchTickets(String apiKey, int company, int contact, int vendor, int product, int ticketOpenStatus, int ticketOwner, String ticketReference, String clientReference, String supplierReference, String companyReference, String dateFrom, String dateTo, int ticketStatus, int priority, String freeText, int pageSize, int pageNum)`

### Parameters

<b>String apiKey</b>	The API identifier key used to identify the user and contract
<b>int company</b>	The company to search tickets for (0 will return tickets for all companies)
<b>int contact</b>	The contact to search tickets for (0 will return tickets for all contacts)
<b>int vendor</b>	The vendor to search tickets for (0 will return tickets for all vendors)
<b>int product</b>	The product to search tickets for (0 will return tickets for all products)
<b>int ticketOpenStatus</b>	The status to of tickets to return (1 = open, 2 = closed, 0 = both)
<b>String ticketReference</b>	The ticket reference to search tickets for ("" will return tickets for all references)
<b>String clientReference</b>	The client reference to search tickets for ("" will return tickets for all references)
<b>String supplierReference</b>	The supplier reference to search tickets for ("" will return tickets for all references)
<b>String companyReference</b>	The company reference to search tickets for ("" will return tickets for all references)
<b>String dateFrom</b>	The date to restrict ticket to being after (dd/MM/yyyy). This searches on the ticket creation date.
<b>String dateTo</b>	The date to restrict ticket to being before (dd/MM/yyyy)
<b>int ticketStatus</b>	The ticket status to search tickets for (0 will return tickets for all ticket statuses)
<b>String freeText</b>	Searches for the presence of the free text within the tickets
<b>int pageSize</b>	The number of records you wish returned (and multiplier for pageNum to get the starting record)
<b>int pageNum</b>	The page you wish returned, pageNum*pageSize gives the first record

**Returns**

An XML string of the tickets that match the search criteria.

**Sample Response**

```
<?xml version="1.0" encoding="UTF-8"?>
<return error="0">
<tickets>
<ticket id="335517" errorType="2030">
<date>2008-11-25 12:12:02.544</date>
<device></device>
<product>Acme Product</product>
<priority>P4 Non-Critical</priority>
<user>Jan Holbourne</user>
<scheddate>2008-11-25 12:12:02.544</scheddate>
<status>Pending Bugfix</status>
</ticket>
<ticket id="333259" errorType="2009">
<date>2008-10-29 15:51:54.005</date>
<device></device>
<product>Acme Product</product>
<priority>P2 Severe Disruption</priority>
<user>Jan Holman</user>
<scheddate>2008-10-29 15:51:54.005</scheddate>
<status>Pending Bugfix</status>
</ticket>
<ticket id="338154" errorType="1116">
<date>2009-01-05 12:07:01.841</date>
<device></device>
<product>Managed-Enquiry</product>
<priority>P4 Non-Critical</priority>
<user>Mark Smithon</user>
<scheddate>2009-01-05 12:07:01.841</scheddate>
<status>Pending Development</status>
</ticket>
</tickets>
</return>
```

**Implementation Notes**

The Company and Contact values should be set to the codes obtained by getCompanies, getContacts respectively. The Product and Vendor parameter values should be set to the product and vendor ids returned by getProducts and getVendors functions respectively.

---

## 5.5. addReminderToTicket Function

Adds a reminder to a ticket.

**Definition** `TicketService.addReminderToTicket(String apiKey, int ticket, String reason, String dateTime)`

**Parameters**

**String apiKey** The API identifier key used to identify the user and contract

**int ticket** The ticket to add the reminder to.

**String reason** The reason for the reminder

**String dateTime** The date and time for the reminder to be set to (dd/MM/yyyy HH:mm)

**Returns** A String "Successfully added reminder" or another string representing the error that occurred

**Sample Response**

```
<?xml version="1.0" encoding="UTF-8"?>
<return error="0">Success</return>
```

---

## 5.6. toggleNotificationForTicket Function

Sets the notification flag on a ticket.

<b>Definition</b>	<b>TicketService.toggleNotificationForTicket(String apiKey, int ticket, boolean notify)</b>
<b>Parameters</b>	
<b>String apiKey</b>	The API identifier key used to identify the user and contract
<b>int ticket</b>	The ticket to toggle notifications for
<b>boolean notify</b>	The new value to set for notifications for this ticket
<b>Returns</b>	A String "Successfully set notification status" or another string representing the error that occurred
<b>Sample Response</b>	<pre>&lt;?xml version="1.0" encoding="UTF-8"?&gt; &lt;return error="0"&gt;Success&lt;/return&gt;</pre>

### ***Implementation Notes***

Disabling the notification flag for a ticket will only ensure that no notifications are sent to the customer contact that are generated from this ticket. To prevent notifications from future tickets for this customer contact set the flags on the Contact or Company accordingly.

---

## 5.7. sendEmailResponseFromTicket Function

Sends an email response from a ticket, adding the email text as an event

**Definition** `TicketService.sendEmailResponseFromTicket(String apiKey, int ticket, int eventType, int secondsSpent, String to, String cc, String bcc, String subject, String messageBody)`

### Parameters

<b>String apiKey</b>	The API identifier key used to identify the user and contract
<b>int ticket</b>	The ticket to send the email from
<b>int eventType</b>	The ticket to send the email from
<b>int secondsSpent</b>	The ticket to send the email from
<b>String to</b>	The email address to send the email to ("" will send to contact for ticket)
<b>String cc</b>	The email address to cc the email to
<b>String bcc</b>	The email address to bcc the email to
<b>String subject</b>	The subject for the email (the string [REF#] will be replaced with the actual reference string)
<b>String messageBody</b>	The content for the email

**Returns** A String "Successfully sent email" or another string representing the error that occurred

**Sample Response**  
<?xml version="1.0" encoding="UTF-8"?>  
<return error="0">Success</return>

## 5.8. getEventsForTicket

Returns all the events listed for a requested ticket

<b>Definition</b>	<b>TicketService.getEventsForTicket(String apiKey, int ticket)</b>
<b>Parameters</b>	
<b>String apiKey</b>	The API identifier key used to identify the user and contract
<b>int ticket</b>	The ticket to send the email from
<b>Returns</b>	An XML string of the events for the given ticket
<b>Sample Response</b>	<pre> &lt;?xml version="1.0" encoding="UTF-8"?&gt; &lt;return error="0"&gt; &lt;ticketdetails&gt;   &lt;ticket id="100909"&gt;     &lt;summary&gt;Testing installation&lt;/summary&gt;     &lt;errorType&gt;General Enquiry&lt;/errorType&gt;     &lt;status&gt;Under Investigation&lt;/status&gt;     &lt;priority&gt;P3&lt;/priority&gt;     &lt;events&gt;       &lt;event id="33"&gt;         &lt;date&gt;02/23/2009 18:14&lt;/date&gt;         &lt;type&gt;Ticket Created&lt;/type&gt;         &lt;user&gt;MRE&lt;/user&gt;         &lt;description&gt;Testing installation&lt;/description&gt;       &lt;/event&gt;       &lt;event id="149"&gt;         &lt;date&gt;03/10/2009 16:57&lt;/date&gt;         &lt;type&gt;Upload&lt;/type&gt;         &lt;user&gt;MF&lt;/user&gt;         &lt;description&gt;File test.txt has been uploaded.&lt;/description&gt;       &lt;/event&gt;       &lt;event id="150"&gt;         &lt;date&gt;03/10/2009 17:09&lt;/date&gt;         &lt;type&gt;Upload&lt;/type&gt;         &lt;user&gt;MF&lt;/user&gt;         &lt;description&gt;File test.txt has been uploaded.&lt;/description&gt;       &lt;/event&gt;     &lt;/events&gt;   &lt;/ticket&gt; &lt;/ticketdetails&gt; &lt;/return&gt; </pre>



---

## 5.9. getAllTags Function

Returns all of the ticket tags for the current contract

<b>Definition</b>	<b>TicketService.getAllTags(String apiKey)</b>
<b>Parameters</b>	
<b>String apiKey</b>	The API identifier key used to identify the user and contract
<b>Returns</b>	An XML string of the Ticket tags for the current contract
<b>Sample Response</b>	<pre>&lt;?xml version="1.0" encoding="UTF-8"?&gt; &lt;tags&gt;   &lt;tag id="1"&gt;Password Reset&lt;/tag&gt;   &lt;tag id="2"&gt;Important&lt;/tag&gt;   &lt;tag id="3"&gt;New User&lt;/tag&gt;   &lt;tag id="5"&gt;Version 4.1&lt;/tag&gt; &lt;/tags&gt;</pre>

---

## 5.10. getClosedTickets

Returns a list of the closed tickets for a give company

<b>Definition</b>	<b>TicketService.getClosedTickets(String apiKey, int iCompany, int pageSize, int pageNum)</b>
<b>Parameters</b>	
<b>String apiKey</b>	The API identifier key used to identify the user and contract
<b>int iCompany</b>	The company that the tickets should belong to
<b>int pageSize</b>	The number of results that should be returned (used for pagination)
<b>int pageNum</b>	The page of results that should be returned
<b>Returns</b>	An XML string of the closed Ticket for the specified company
<b>Sample Response</b>	<pre>&lt;?xml version="1.0" encoding="UTF-8"?&gt; &lt;tickets ticketCount="124"&gt;   &lt;ticket id="100900"&gt;     &lt;summary&gt;System is slow&lt;/summary&gt;     &lt;date&gt;100123489741&lt;/date&gt;     &lt;dateclosed&gt;10012348979&lt;/dateclosed&gt;     &lt;issue&gt;User has reported that connection time to the system is slow&lt;/issue&gt;     &lt;contact&gt;John Smith&lt;/contact&gt;     &lt;status&gt;Closed&lt;/status&gt;   &lt;/ticket&gt; &lt;/tickets&gt;</pre>

---

## 5.11. getOpenTickets Function

Returns a list of the open tickets for a give company

**Definition** `TicketService.getOpenTickets(String apiKey, int iCompany, int iChangeType, int iSupportType)`

**Parameters**

**String apiKey** The API identifier key used to identify the user and contract

**int iCompany** The company that the tickets should belong to

**int iChangeType**

**int iSupportType**

**Returns** An XML string of the open Ticket for the specified company

**Sample Response**

```
<?xml version="1.0" encoding="UTF-8"?>
<tickets ticketCount="124">
  <ticket id="100900" errorType="15">
    <created>2010-03-14 15:45:00</created>
    <device>REF-01221AA</device>
    <product>IssueCentre 2.4.1</Product>
    <priority>P3</priority>
    <contact>John Smith</contact>
    <status>Under Investigation</status>
  </ticket>
</tickets>
```

---

## 5.12.getTicketFixGroup Function

Returns a list of the fix groups that apply to a given ticket

**Definition** `TicketService.getOpenTickets(String apiKey, int iTicket)`

**Parameters**

**String apiKey** The API identifier key used to identify the user and contract  
**int iTicket** The ticket that the Fix Groups should apply to

**Returns** An XML string of the Fix Groups that apply to the Ticket

**Sample Response**

```
<?xml version="1.0" encoding="UTF-8"?>
<fixgroups>
  <fixgroup icode="12">Front Line</fixgroup>
  <fixgroup icode="15">Second Line</fixgroup>
  <fixgroup icode="19">Management</fixgroup>
</fixgroups>
```

---

## 5.13. setTicketFixGroup Function

Returns a list of the closed tickets for a give company

**Definition** `TicketService.getClosedTickets(String apiKey, int iTicket, int iFixGroup)`

**Parameters**

**String apiKey** The API identifier key used to identify the user and contract

**int iTicket** The Ticket that is being updated

**int iFixGroup** The fix group that the Ticket will belong to

**Returns** An XML string of the closed Ticket for the specified company

**Sample Response**

```
<?xml version="1.0" encoding="UTF-8"?>
<return error="0">100901</return>
```